System Verification and Validation Plan Checklist

Spencer Smith

October 11, 2025

 Follows writing checklist (full checklist provided in a separate document) 	
□ LATEX points	
□ Structure	
☐ Spelling, grammar, attention to detail	
☐ Avoid low information content phrases	
☐ Writing style	
• Follows the template, all parts present	
☐ Table of contents	
☐ Pages are numbered	
\square Revision history included for major revisions	
\square Sections from template are all present	
□ Values of auxiliary constants are given (constants are used to improve maintainability and to increase understandability)	
• Grammar, spelling, presentation	
\square No spelling mistakes (use a spell checker!)	
$\hfill \square$ No grammar mistakes (review, ask someone else to review (at least a few sections))	
$\hfill\Box$ Paragraphs are structured well (clear topic sentence, cohesive)	

□ Paragraphs are concise (not wordy)
$\hfill\square$ No Low Information Content (LIC) phrases (List of LIC phrases)
☐ All hyperlinks work
☐ Every figure has a caption
☐ Every table has a heading
$\hfill\square$ Symbolic names are used for quantities, rather than literal values
• LaTeX
☐ Template comments do not show in the pdf version, either by removing them, or by turning them off.
$\hfill\square$ References and labels are used so that maintenance is feasible
• Overall qualities of documentation
\square Test cases include SPECIFIC input
\square Test cases include EXPLICIT output
\square Description over specification, when appropriate
□ Plans for what to do with description data (performance, usability, etc). This may involve saying what plots will be generated.
\Box Plans to quantify error for scalar values using relative error
\square Plans to quantify error for vector and matrix values using a norm of an error vector (matrix)
$\hfill\square$ Plans are feasible (can be accomplished with resources available)
\square Plans are ambitious enough for an A+ effort
\square Survey questions for usability survey are in an Appendix (if appropriate)
\square Plans for task based inspection, if appropriate
\square Very careful use of random testing
☐ Specific programming language is listed
\square Specific linter tool is listed (if appropriate)
\square Specific coding standard is given
\square Specific unit testing framework is given

Investigation of code coverage measuring tools
Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
Specific performance measuring tools listed (like Valgrind), if appropriate
Traceability between test cases and requirements is summarized (likely in a table)